WHAT IS CLAIMED:

1. A system comprising:

a central processing unit operating in response to a set of instructions for processing information;

an interface for providing access to selected circuitry forming a part of said system on a chip by an external device; and

a set of non-volatile programmable security elements for selectively enabling and disabling the operation of said interface to provide a private environment for processing said information.

2. The system of Claim 1 wherein said interface comprises a JTAG port.

3. The system of Claim 1 wherein said interface comprises an in-circuit emulation port.

4. The system of Claim 1 wherein said interface comprises a port allowing said external device to observe an internal state of said integrated circuit.

5. The system of Claim 1 and further comprising boot memory for storing security initialization code, said security initialization code selectively enabled by programming said set of programmable elements.

6. The system of Claim 5 and further comprising boot memory for storing security initialization code, said security initialization code selectively enabled by programming said set of programmable elements.

7. The system of Claim 1 and further comprising a cache associated with said central processing unit, said cache including a selected number of lockable entries for storing secure information.

8. The system of Claim 1 and further comprising a translation look aside buffer, with said CPU, said translation look aside buffer including a selected number of lockable entries for storing addresses to secure information in memory

9. The system of Claim 1 and further comprising on-chip random access memory including a selected amount of memory space for storing address translation tables.

10. The system of Claim 1 wherein said set of programmable elements comprises a set of fuses.

11. The system of Claim 1 wherein said set of programmable elements comprise a set of bond options.

12. The system of Claim 1 wherein said set of programmable elements comprises a set of antifuses.

13. The system of Claim 1 wherein said set of programmable elements comprises a set of read-only memory cells.

14. The system of Claim 1 wherein said set of programmable elements comprises a set of write-once memory cells.

15. The system of Claim 1 wherein said set of programmable elements comprises a set of FLASH memory cells.

16. A method for selective secure operation of a system comprising the steps of:

disabling debug circuitry forming a part of the system on a chip at power on reset to prevent access by an unauthorized party to security resources;

determining whether a security procedure is called for during system initialization from boot memory; and

attempting to execute a selected security procedure when a security procedure is called for during system initialization, comprising the steps of:

mapping a vector to the selected location in boot memory storing security code calling a selected security procedure;

executing the security code in boot memory to determine whether the called security procedure is valid; and

operating the system in a secure environment in response to the called security procedure when the called security procedure is valid.

17. The method of Claim 16 and further comprising the step of booting for unsecured operation if a security procedure is not called for in response to said step of determining, comprising the substeps of:

selecting one of internal AND external boot memory options;

in response to the selection of the internal boot memory option, mapping the vector to a default location in internal boot memory;

in response to the selection of the external boot memory option, mapping the vector to a location in external boot memory;

enabling the debug circuitry and

executing boot code pointed-to by the vector.

18. The method of Claim 17 wherein said substep of mapping the vector in external memory comprises the further substeps of:

remapping a chip select signal controlling the external memory to point-to currently executing memory space; and

changing a program counter to the vector such that a fetch of an instruction changing the program counter is completed prior to completion of said step of remapping.

19. The method of Claim 18 wherein said vector comprises a CPU reset vector.

20. The method of Claim 16 wherein execution of the security code in boot memory determines that the called security procedure is invalid and said method further comprises the steps of:

remapping the vector to the boot memory to a location storing second selected security code, the second security code calling a second security procedure;

executing the second selected security code in boot memory to determine if the second security procedure is valid; and

operating the system on a chip in a secure environment in response to the second security procedure when the second security procedure is valid.

21. The method of Claim 16 wherein said step of executing the security code in boot memory to determine whether the called security procedure is valid comprises the substep of searching for the called security procedure in external memory coupled to the system on a chip.

22. The method of Claim 16 and further comprising the step of executing default boot code when the called security procedure is invalid.

23. The method of Claim 16 wherein said step of determining if a security procedure is called for during system initialization comprises the substep of reading the state of a set of programmable elements.

1 24. The method of Claim 23 wherein said substep of reading
2 is performed by logic gates.

1 25. The method of Claim 23 wherein said substep of reading
2 is performed by a central processing unit

26. A method of preventing access and observation of encached information comprising the steps of:

generating private information to be encached;

storing the private information in memory;

updating a translation look aside buffer with descriptors to locations in memory containing the private information;

forcing a cache miss to a selected location in cache to be loaded with a selected portion of the private information;

retrieving the selected portion of the private information from memory using a corresponding descriptor from the translation look aside buffer;

loading the retrieved portion of the private information into the selected location in cache; and

locking the selected portion of the private information in the selected location in cache.

27. The method of Claim 26 and further comprising the step of locking the descriptor corresponding to the selected portion of the private information in the translation look aside buffer.

28. The method of Claim 26 wherein said selected location in cache is associated with a replacement counter base and said step of locking comprises the substep of resetting the replacement counter base to a value higher than the replacement counter base associated with the selected location in cache.

29. The method of Claim 26 wherein said step of updating the translation look aside buffer comprises the substeps of:

setting up a translation table including entries for generating the descriptors to memory locations storing the private information;

updating a replacement counter to point to a current translation look aside buffer entry to be filled;

forcing a miss to the current translation look aside buffer entry;

performing a table walk through the translation table to generate a descriptor associated with private information in memory; and

loading the descriptor obtained from the table walk in the current translation look aside buffer entry.

30. The method of Claim 26 wherein said step of loading the selected portion of the decoded information in cache comprises the step of loading a cache line in instruction cache.

31. The method of Claim 26 wherein said step of loading the selected portion of the private information in cache comprises the step of loading a cache line in data cache.

32. The method of Claim 26 wherein said step of setting up a translation table comprises the step of setting up an emulated translation table.

33. A method of synthesizing translation tables comprising the steps of:

setting up at least one register for storing information controlling access to a plurality of memory spaces;

generating a virtual address including a pointer to selected information in the at least one register controlling access to a selected one of the memory spaces;

accessing said selected information at said pointer from the at least one register; and

generating a physical address to the selected one of the memory spaces from the information accessed from the at least one register.

34. The method of Claim 33 wherein the selected information comprises access permissions.

35. The method of Claim 33 wherein the selected information comprises cacheability and bufferability bits.

36. The method Claim 33 wherein the at least one register comprises a first register for storing access permissions associated with each of the memory spaces, a second register for storing a cacheability bit associated with each of the memory spaces and a third register for storing a bufferability bit associated with each of the memory spaces.

37. The method of Claim 33 wherein the selected information accessed from the at least one register comprises a base address to at least one second level register controlling access to a selected part of a selected one of the memory spaces and said step of generating a physical address comprises the substeps of:

accessing selected information in the at least one second level register using the base address and a table index from the virtual address; and

generating the physical address from the selected information accessed from the at least one second level register and page index bits from the virtual address.

38. The method of Claim 33 wherein said information includes for each of the memory spaces a pair of access permission bits, a bufferability bit and a cacheability bit.

39. A method of performing an emulated translation table walk comprising the steps of:

emulating a translation register including a plurality of entries populated with descriptors;

emulating an index register storing indices associated with the entries of the emulated translation register;

pointing to the emulated translation register with a translation base pointer;

generating an address including index bits to the emulated translation register;

comparing the index bits from the address with the indices stored in the index register; and

selectively accessing a corresponding descriptor in the translation table in response to said step of comparing.

40. The method of Claim 39 wherein said step of generating an address comprises the step of generating a virtual address forcing a miss to an associated cache.

41. The method of Claim 39 wherein the descriptors comprise selected physical address bits and access permissions and said method further comprises the steps of:

determining from the permissions from the descriptor selectively accessed from the emulated translation table whether a corresponding access to memory is allowed; and

if the access is allowed, generating a physical address to a location in memory using the physical address bits from the accessed descriptor.

42. The method of Claim 39 and further comprising the steps of:

populating selected entries of the translation register with second level base addresses;

emulating a second level translation register including a plurality of entries populated with second level descriptors;

emulating a second level index register populated with indices to corresponding entries in the second level translation register;

accessing the second level translation register with a base address from the translation register;

comparing index bits from the virtual address with indices in the second level index register; and

in response to said step of comparing, selectively accessing a second level descriptor from the corresponding entry in the second level translation table.

43. The method of Claim 42 wherein the second level descriptors comprise selected second level physical address bits and second level access permissions and said method further comprises the steps of:

determining from the second level permissions of the descriptor selectively accessed from the emulated second level translation table whether a corresponding access to memory is allowed; and

if the access is allowed, generating a physical address to a location in memory using the second level physical address bits from the accessed second level descriptor.

44. A system comprising:

a central processing unit operating in response to a set of instructions for processing information;

an interface for providing access to selected circuitry forming a part of said system on a chip by an external device; and

a set of programmable security elements for selectively enabling and disabling the operation of said interface to provide a private environment for processing said information.

45. The system of Claim 44 wherein said central processing unit, said interface, and said security elements are fabricated on a single integrated circuit chip.

46. The system of Claim 45 wherein said integrated circuit chip further includes on-chip read-only memory.

47. The system of Claim 45 wherein said integrated circuit chip further includes on-chip random access memory.

48. The system of Claim 44 and further comprising memory storing private code for initializing private operation of said system.

49. The system of Claim 44 wherein said system forms a portion of a hand-held personal appliance.

50. The system of Claim 49 wherein said hand-held appliance comprises and audio decoder.

51. A hand-held audio decoder comprising:

a central processing unit operating in response to a set of instructions for decoding a stream of encoded digital audio data;

memory for storing said set of instructions; and

digital to analog converter circuitry for generating audio from said decoded stream of digital audio data.

52. The audio decoder of Claim 51 wherein said central processing unit comprises an advanced risk machine.

53. The audio decoder of Claim 51 wherein said stream of encoded digital data comprises a stream of MPEGx, Layer 3 encoded audio data.

54. The audio decoder of Claim 51 wherein said stream of encoded digital data comprises a stream of ACC encoded digital data.

55. The audio decoder of Claim 51 wherein said stream of encoded digital data comprises a stream of MS Audio encoded digital data.

56. The audio decoder of Claim 51 wherein said decoder is capable of operating correctly from one AA battery for a period of at least one hour.

57. A method of synthesized address translation comprising the steps of:

setting up at least one global register having a plurality of entries each for storing access control bits for a corresponding region of memory each comprising a plurality of locations having common access characteristics; and

setting up an individual register for storing a descriptor corresponding to a region of memory having differing access characteristics;

generating an address including an index;

in response to a first state of the index, accessing said descriptor from the individual register; and

in response to a second state of the index, performing the substeps of:

accessing the access control bits from a selected one of the global registers pointed-to by said index; and

generating a descriptor by merging the access control bits accessed from the selected one of the global registers with selected bits of said address.

58. The method of Claim 57 and further comprising the steps of:

setting up a constant register for storing a constant; and

in response to a third state of the index, accessing a constant from said constant register.

59. The method of Claim 58 wherein the constant register comprises hardwired gates.

60. The method of Claim 57 wherein the access control bits comprise access permission bits, cacheability bits and bufferability bits.

61. The method of Claim 57 wherein said at least one register comprises a first register having a plurality of entries each for storing access permission bits for a corresponding one of the regions, a second register having a plurality of entries each for storing a cacheability bit for a corresponding one of the regions and a third register having a plurality of entries each for storing a bufferability bit for a corresponding one of the regions.

62. The method of Claim 57 wherein said descriptor comprises first level descriptor including an second level index, the method further comprising the steps of:

setting up at least one second level global register having a plurality of entries each for storing access control bits for a corresponding region of memory comprising a plurality of locations having common access characteristics; and

setting up a second level individual register for storing a descriptor corresponding to a region of memory having differing access characteristics;

in response to a first state of the second level index, accessing the descriptor from the second level individual register; and

in response to a second state of the second level index, performing the substeps of:

accessing said access control bits from a selected one of the second level global registers pointed-to by said second level index; and

generating a second descriptor by merging the access control bits accessed from the selected one of the second level global registers with selected bits of the address.

63. The method of Claim 62 and further comprising the steps of:

setting up a second level constant register for storing a second level constant; and

in response to a third state of the second level index, accessing a second level constant from the second level constant register.

64. The method of Claim 58 wherein the second constant register comprises hardwired gates.

65. The method of Claim 57 wherein the access control bits comprise access permission bits, cacheability bits and bufferability bits.